

Fast Update of Conditional Simulation Ensembles

Clément Chevalier, Xavier Emery, and David Ginsbourger

Abstract Gaussian random field (GRF) conditional simulation is a key ingredient in many spatial statistics problems for computing Monte-Carlo estimators and quantifying uncertainties on non-linear functionals of GRFs conditional on data. Conditional simulations are known to often be computer intensive, especially when appealing to matrix decomposition approaches with a large number of simulation points. This work studies settings where conditioning observations are assimilated batch-sequentially, with one point or a batch of points at each stage. Assuming that conditional simulations have been performed at a previous stage, the goal is to take advantage of already available sample paths and by-products in order to produce updated conditional simulations at minimal cost. Explicit formulae are provided, which allow updating an ensemble of sample paths conditioned on $n \geq 0$ observations to an ensemble conditioned on $n + q$ observations, for arbitrary $q \geq 1$. Compared to direct approaches, the proposed formulae prove to substantially reduce computational complexity. Moreover, these formulae explicitly exhibit how the q new observations are updating the old sample paths. Detailed complexity calculations highlighting the benefits of this approach with respect to state-of-the-art algorithms are provided and are complemented by numerical experiments.

C. Chevalier

Institute of Mathematics, University of Zurich, Winterthurerstrasse 190, CH-8057 Zurich, Switzerland

e-mail: `clement.chevalier@math.uzh.ch`

X. Emery

Mining Engineering Department / Advanced Mining Technology Center, University of Chile, Avenida Tupper 2069, Santiago, Chile

e-mail: `xemery@ing.uchile.cl`

D. Ginsbourger (✉)

IMSV, Department of Mathematics and Statistics, University of Bern, Alpeneggstrasse 22, 3012 Bern, Switzerland

e-mail: `david.ginsbourger@stat.unibe.ch`

Keywords Gaussian random fields · residual kriging algorithm · batch-sequential strategies · kriging update equations

1 Introduction

Throughout the paper, $Z = (Z(\mathbf{x}))_{\mathbf{x} \in \mathbb{X}}$ is a random field defined on a probability space (Ω, \mathcal{B}, P) , with index \mathbf{x} lying in a bounded set $\mathbb{X} \subset \mathbb{R}^d$ ($d \geq 1$). The random field Z is assumed to be evaluated sequentially, first at n points $\mathbf{x}_{1:n} := (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathbb{X}^n$ ($n \geq 0$), and then at q additional points $\mathbf{x}_{n+(1:q)} := (\mathbf{x}_{n+1}, \dots, \mathbf{x}_{n+q}) \in \mathbb{X}^q$ ($q \geq 1$). A crucial assumption here concerning the random field Z is that its distribution at step n be Gaussian. This includes of course the case when Z is a GRF, but also the case of intrinsic random fields with Gaussian generalized increments (Matheron, 1973) and the Bayesian settings (Omre and Halvorsen, 1989; Handcock and Stein, 1993) where Z is Gaussian conditionally on some linear trend parameters with improper uniform distribution, and $n \geq 1$ pointwise evaluations of Z are already available at step n .

Assuming that J simulations of Z have been performed at stage n , this work details procedures to update them when a vector of new observations $Z(\mathbf{x}_{n+(1:q)})$ is assimilated. More precisely, the goal is to get a fast algorithm that generates J sample paths, rigorously drawn from the distribution of Z conditional on all $n+q$ evaluations, by recycling previous simulations and calculations as much as possible.

The main contribution of this paper is illustrated on Fig. 1. An ensemble of 50 simulations of a GRF Z are performed conditionally on $n = 6$ observations (black curves). A fast simulation update procedure (the so-called FOXY algorithm, presented in detail in Section 2) is then applied to this ensemble in order to condition it on $q = 3$ additional observations at points $\mathbf{x}_{n+(1:q)} = (0.25, 0.3, 0.8)$, yielding the red curves. Z is here assumed to be stationary, centered, and to possess a Matérn covariance kernel with regularity parameter $\nu = 3/2$ (Stein, 1999).

Motivations for developing such a procedure come from problems in various application fields. While GRF conditional simulation constitutes a standard and important topic in the literature of geostatistics (Chilès and Delfiner, 2012) with a variety of applications in geosciences and natural resources characterization (Delhomme, 1979; Deutsch, 2002; Journel and Kyriakidis, 2004; Chilès and Allard, 2005; Dimitrakopoulos, 2011), they have been increasingly used in engineering and related areas, where GRF models have been used as prior distributions on expensive-to-evaluate functions (Hoshiya, 1995; Santner et al., 2003; Villemonteix et al., 2009; Roustant et al., 2012; Binois et al., 2014). For the latter applications, conditional simulations are typically needed to compute expectations of non-linear functionals of random fields. This includes for example the maximum, but also the Shannon entropy of the maximizer (Villemonteix et al., 2009).

As conditional simulation methods are known to be generally computer intensive, an important challenge is the reduction of computation time through efficient

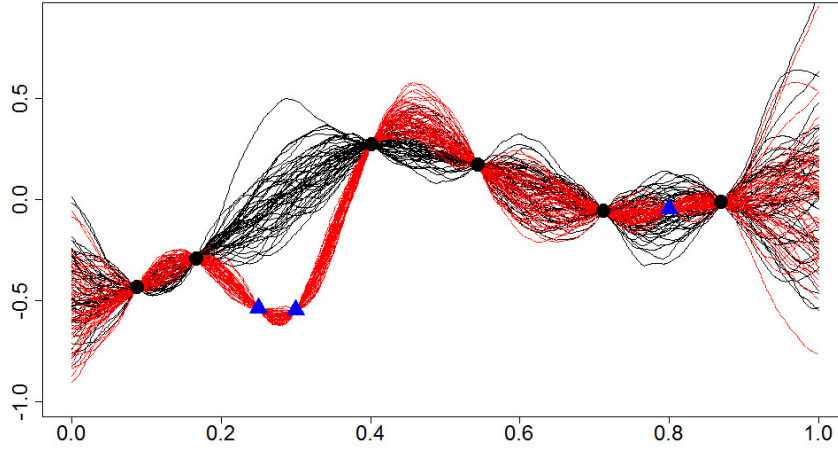


Fig. 1 GRF simulations conditioned on $n = 6$ (black curves) and $n + q = 9$ observations (red curves). Black circles stand for $n = 6$ initial observations and blue triangles represent $q = 3$ additional observations

algorithms. In the frequent case where observations are assimilated sequentially, both new observations and previously simulated sample paths are typically available. It is then tempting to take advantage of the latter for obtaining sample paths conditioned on all observations without having to restart everything from scratch. A well-known conditional simulation algorithm which may apply to these settings is the residual substitution approach (Chilès and Delfiner, 2012; de Fouquet, 1994), also called conditioning kriging, or residual kriging algorithm. This method starts from non-conditional GRF simulations obtained by using algorithms such as the circulant-embedding, spectral methods, or turning bands (Chilès and Delfiner, 2012; Emery and Lantuéjoul, 2006), and the assimilation of data is then achieved via kriging (Hernández and Emery, 2009).

In the present paper, efficient formulae allowing a fast update of GRF sample paths are provided, together with an algorithm, FOXY, the complexity of which is studied in detail to justify the improvement with respect to the residual kriging algorithm. The acronym FOXY stands for fast update of conditional simulation ensembles (FOCSE, alias FOXY). The new formulae have the advantage of analytically exhibiting the dependence between updated GRF sample paths and the newly assimilated observations. One of the key ingredients to obtain the formulae and set up the algorithm happens to be the batch-sequential kriging update formulae of Emery (2009) and Chevalier et al. (2014), as explained next. The paper is organized as follows. Section 2 presents two already well-established approaches, namely the residual kriging algorithm and the kriging update formulae which enable to derive the proposed conditional simulation update formula. Related algorithms and their

complexity are presented in Sect. 3. Subsequent numerical experiments illustrating the efficiency of the FOXY algorithm are finally given in Sect. 4. For brevity and self-containedness, basics of kriging and more detailed versions of the algorithms are given in appendices.

2 Theory: From Residual Kriging to the FOXY Algorithm

This section gives the main result of the paper. Sections. 2.1 and 2.2 present two crucial ingredients which are used to obtain the conditional simulation update formula in Sect. 2.3.

2.1 Residual Kriging Algorithm

In the simple kriging settings, the residual kriging algorithm or residual substitution approach (Chilès and Delfiner, 2012) is known to provide a simple and efficient way to produce simulations of a GRF Z conditional on observations at $\mathbf{x}_{1:q}$ ($q \geq 1$) relying both on non-conditional simulations of Z and simple kriging means of Z and of these non-conditional simulations given their respective values at $\mathbf{x}_{1:q}$. Denoting by $z(\mathbf{x}_i)$ ($1 \leq i \leq q$) the values of Z observed at $\mathbf{x}_{1:q}$, by z' or $z^{(j)}$ ($1 \leq j \leq J$) some non-conditional realizations of Z , and by $\mathbf{E}_p := \{\mathbf{e}_1, \dots, \mathbf{e}_p\} \subset \mathbb{X}$ a considered set of simulation points (now assumed to be a finite subset of \mathbb{X} , for simplicity) the procedure consists in Algorithm 1 below.

Algorithm 1 Standard residual kriging algorithm (case of several replicates)

Require: The distribution of the GRF Z

Require: Evaluation points $\mathbf{x}_{1:q} = (\mathbf{x}_1, \dots, \mathbf{x}_q) \in \mathbb{X}^q$

Require: Evaluation results $z(\mathbf{x}_i)$ ($1 \leq i \leq q$)

Require: Simulation points $\mathbf{E}_p = \{\mathbf{e}_1, \dots, \mathbf{e}_p\} \subset \mathbb{X}$

Step 1. Simulate J replicates of Z at $\{\mathbf{x}_1, \dots, \mathbf{x}_q\} \cup \{\mathbf{e}_1, \dots, \mathbf{e}_p\}$, denoted $z^{(j)}$ ($1 \leq j \leq J$)

Step 2. Calculate the kriging mean function m of Z knowing $z(\mathbf{x}_i)$ ($1 \leq i \leq q$), and evaluate it at the simulation points \mathbf{E}_p , delivering the vector $m(\mathbf{E}_p)$.

Step 3.

for $j = 1 \rightarrow J$ **do**

- Calculate the kriging mean $m^{(j)}$ of Z knowing $z^{(j)}(\mathbf{x}_{1:q})$, and evaluate it at the simulation points \mathbf{E}_p , delivering the vector $m^{(j)}(\mathbf{E}_p)$. The kriging weights are the same as that calculated at Step 2.

- Set $r^{(j)}(\mathbf{E}_p) = z^{(j)}(\mathbf{E}_p) - m^{(j)}(\mathbf{E}_p)$

- Set $\mathbf{z}^{*(j)} = m(\mathbf{E}_p) + r^{(j)}(\mathbf{E}_p)$

end for

Return $\mathbf{z}^{*(1)}, \dots, \mathbf{z}^{*(J)}$ as conditional simulations of Z at \mathbf{E}_p knowing $Z(\mathbf{x}_{1:q}) = z(\mathbf{x}_{1:q})$.

While this algorithm is common knowledge in geosciences, it is usually restricted to the simple and ordinary kriging frameworks and presented rather succinctly across the geostatistics literature (de Fouquet, 1994; ?; Chilès and Delfiner, 2012). This procedure is recalled below.

Proposition 1 (residual kriging algorithm) *Let $q \geq 1$, $\mathbf{x}_{1:q} = (\mathbf{x}_1, \dots, \mathbf{x}_q) \in \mathbb{X}^q$, Z be a GRF with conditional expectation $M_q := \mathbb{E}(Z|Z(\mathbf{x}_{1:q}))$, and Z' be a replicate of Z with $M'_q := \mathbb{E}(Z'|Z'(\mathbf{x}_{1:q}))$. Then, $Z^* = M_q + Z' - M'_q$ is equal to Z in distribution, so that conditional on the event $Z(\mathbf{x}_{1:q}) = \mathbf{z}_{1:q}$ (for arbitrary $\mathbf{z}_{1:q} \in \mathbb{R}^q$) and denoting by m_q the corresponding realization of M_q*

$$(Z|Z(\mathbf{x}_{1:q}) = \mathbf{z}_{1:q}) \stackrel{\mathcal{D}}{=} m_q + Z' - M'_q. \quad (1)$$

Furthermore, if $Z^{(1)}, \dots, Z^{(J)}$ ($J \geq 1$) are independent replicates of Z with $M_q^{(j)} := \mathbb{E}(Z^{(j)}|Z^{(j)}(\mathbf{x}_{1:q}))$ ($1 \leq j \leq J$), the random fields $m_q + Z^{(j)} - M_q^{(j)}$ are stochastically independent replicates of Z conditionally on $Z(\mathbf{x}_{1:q}) = \mathbf{z}_{1:q}$.

Proof. By Gaussianity of Z , the following decomposition holds

$$Z = M_q + R_q, \quad (2)$$

where $M_q = \mathbb{E}(Z|Z(\mathbf{x}_{1:q}))$ depends on Z only through its values at $\mathbf{x}_{1:q}$ (technically, M_q is $\sigma(Z(\mathbf{x}_{1:q}))$ -measurable) and $R_q := (Z - M_q)$ is independent of $Z(\mathbf{x}_{1:q})$. The same straightforwardly applies to the replicates $Z^{(1)}, \dots, Z^{(J)}$, and the use of similar notations gives $Z^{(j)} = M_q^{(j)} + R_q^{(j)}$, where the $M_q^{(j)}$'s are $\sigma(Z^{(j)}(\mathbf{x}_{1:q}))$ -measurable and the $R_q^{(j)}$'s are respectively independent of $Z^{(j)}(\mathbf{x}_{1:q})$ ($1 \leq j \leq J$). Defining

$$Z^{*(j)} = M_q + Z^{(j)} - M_q^{(j)} = M_q + R_q^{(j)} \quad (1 \leq j \leq J)$$

and using the fact that $R_q^{(j)}$ and R_q have same distribution and are both independent of M_q , one easily obtains that $Z^{*(j)} \stackrel{\mathcal{D}}{=} Z$. Besides, by independence between the residuals and the values of the respective random fields at points $\mathbf{x}_{1:q}$

$$(Z^{*(j)}|Z(\mathbf{x}_{1:q}) = \mathbf{z}_{1:q}) \stackrel{\mathcal{D}}{=} (m_q + R_q^{(j)}) \stackrel{\mathcal{D}}{=} (m_q + R_q) \stackrel{\mathcal{D}}{=} (Z|Z(\mathbf{x}_{1:q}) = \mathbf{z}_{1:q}). \quad (3)$$

Finally, the $Z^{*(j)}$'s are indeed independent conditionally on $Z(\mathbf{x}_{1:q})$ by independence of the $R_q^{(j)}$'s. \square

An example of application of Proposition 1 in a non-standard set up is given in Fig. 2. A universal kriging model is assumed, in which Z has already been evaluated at n (not represented) points prior to the evaluation at the q conditioning points, so that Z 's distribution at stage n is indeed Gaussian but with non-stationary mean and covariance kernel given by the universal kriging equations (recalled in Appendix A).

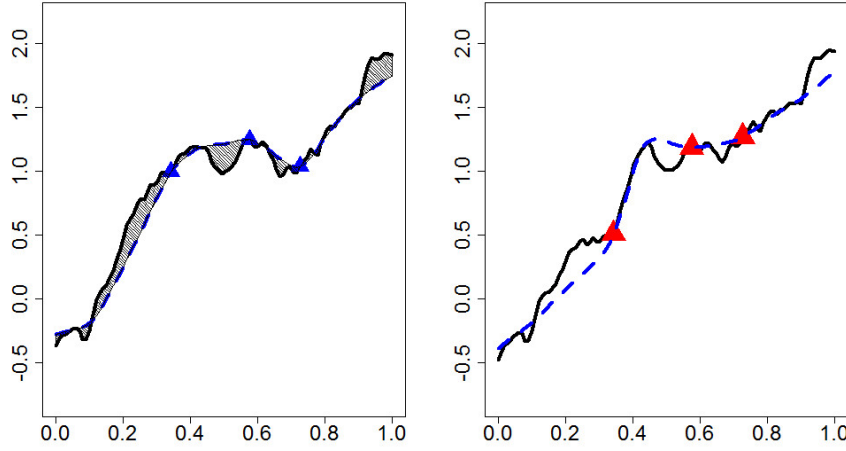


Fig. 2 Left: kriging residual obtained from a non-conditional simulation of a GRF $Z^{(j)}$ (black solid line) and its simple kriging mean (blue dashed line) based on $q = 3$ observations (blue triangles). Right: conditional simulation of Z (black solid line) obtained by summing Z 's simple kriging mean (blue dashed line) based on its values at the same design (red triangles) and the kriging residual simulated on the left graph

From that perspective, the conditional expectation of Z when $q = 3$ new observations are available can be obtained using simple kriging with the previous non-stationary mean and covariance kernel. Let us stress here that computing simple kriging means with a non-conditional covariance function being the covariance function of the universal kriging errors (referred to as the universal kriging covariance function) based on n past observations is not necessarily sensible from a computational point of view, but that having this particular approach in mind will facilitate understanding forthcoming ideas. Such non-stationary simple kriging mean actually coincides with the usual universal kriging mean relying on all $n + q$ observations, as can be seen for instance through the update equations recalled in the next section.

2.2 Kriging Update Framework

Keeping in mind the overall set up of a random field whose distribution at stage $n \geq 0$ is Gaussian with mean M_n (the realization of which is denoted m_n) and covariance k_n , let us now focus on the situation where a batch of evaluations at $q > 0$ additional points $\mathbf{x}_{n+(1:q)} = (\mathbf{x}_{n+1}, \dots, \mathbf{x}_{n+q})$ is assimilated.

In the literature, a lot of efforts have been paid to obtain update formulae for kriging predictors when observations are assimilated sequentially. Update formulae

are meant to enable a fast computation of the kriging mean M_{n+q} and the kriging covariance function k_{n+q} in case M_n and k_n are already available. Barnes and Watson (1992) gave kriging update formulae for the kriging mean and covariance function in simple kriging settings, with $q = 1$. Gao et al. (1996) generalized these formulae, still with $q = 1$, to universal kriging settings. Finally Emery (2009) obtained universal kriging update formulae for the kriging mean for arbitrary $q \geq 1$ and Chevalier et al. (2014) complemented them with update formulae for the kriging covariance function. These formulae, adapted to settings and notations of this article are recalled below

$$M_{n+q}(\mathbf{x}) = M_n(\mathbf{x}) + \lambda_{n,q}(\mathbf{x})^\top (Z(\mathbf{x}_{n+(1:q)}) - M_n(\mathbf{x}_{n+(1:q)})), \quad (4)$$

$$k_{n+q}(\mathbf{x}, \mathbf{x}') = k_n(\mathbf{x}, \mathbf{x}') - \lambda_{n,q}(\mathbf{x})^\top K_{n,q} \lambda_{n,q}(\mathbf{x}'), \quad (5)$$

$$\lambda_{n,q}(\mathbf{x}) = K_{n,q}^{-1} k_n(\mathbf{x}, \mathbf{x}_{n+(1:q)}), \quad (6)$$

where $\lambda_{n,q}(\mathbf{x})$ is a vector of q kriging weights of responses at $\mathbf{x}_{n+(1:q)}$ for predicting at point \mathbf{x} at time n , and $K_{n,q} := (k_n(\mathbf{x}_{n+i}, \mathbf{x}_{n+j}))_{1 \leq i, j \leq q}$. These formulae enable significant computational savings as only q kriging weights $\lambda_{n,q}$ (and not $n+q$) need to be computed for obtaining updated kriging mean and covariance functions. In particular, a cumbersome $(n+q) \times (n+q)$ matrix inversion is avoided.

Note that the kriging update formulae (4), (5) corroborate the fact mentioned in the previous example (Fig. 2). Performing simple kriging using m_n and k_n as mean and covariance functions and q new observations actually gives the same result as performing universal kriging based on the initial model, with $n+q$ observations.

2.3 FOXY: Fast Update of Conditional Simulation Ensembles

All the ingredients are now gathered to detail a new update formula which will serve as a basis to a method for fast updating ensembles of conditional simulations, referred to as the FOXY algorithm. Let us now state the main result of the paper, all notations being kept as in the previous sections unless precised otherwise.

Proposition 2 (Conditional simulation update formula) *Let $Z^{(1)}, \dots, Z^{(J)}$ be independent replicates of Z conditional on $Z(\mathbf{x}_{1:n})$. Then, the random fields*

$$Z^{*(j)} := Z^{(j)} + \lambda_{n,q}^\top (Z(\mathbf{x}_{n+(1:q)}) - Z^{(j)}(\mathbf{x}_{n+(1:q)})) \quad (j \in \{1, \dots, J\}) \quad (7)$$

have the same distribution as Z conditioned on $Z(\mathbf{x}_{1:(n+q)})$. In other words, for any conditioning values $\mathbf{z}_{1:n} \in \mathbb{R}^n$, $\mathbf{z}_{n+(1:q)} \in \mathbb{R}^q$

$$Z | (Z(\mathbf{x}_{1:n}) = \mathbf{z}_{1:n}, Z(\mathbf{x}_{n+(1:q)}) = \mathbf{z}_{n+(1:q)}) \stackrel{\mathcal{D}}{=} Z^{(j)} + \lambda_{n,q}^\top (\mathbf{z}_{n+(1:q)} - Z^{(j)}(\mathbf{x}_{n+(1:q)})). \quad (8)$$

Furthermore, the $Z^{(j)} + \lambda_{n,q}^\top (\mathbf{z}_{n+(1:q)} - Z^{(j)}(\mathbf{x}_{n+(1:q)}))$'s are stochastically independent.

Proof. Let $M_{n+q} := \mathbb{E}(Z|Z(\mathbf{x}_{1:n}), Z(\mathbf{x}_{n+(1:q)}))$ and $M_{n+q}^{(j)}$ be defined similarly for the j^{th} replicate $Z^{(j)}$. The equality in distribution of Z and $Z^{(j)}$ implies that

$$Z - M_{n+q} \stackrel{\mathcal{D}}{=} Z^{(j)} - M_{n+q}^{(j)}. \quad (9)$$

Now, an application of the kriging update formula (4) to both M_{n+q} and $M_{n+q}^{(j)}$, together with the identity $M_n = M_n^{(j)}$, yields

$$\begin{aligned} Z &\stackrel{\mathcal{D}}{=} M_n + \lambda_{n,q}^\top (Z(\mathbf{x}_{n+(1:q)}) - M_n(\mathbf{x}_{n+(1:q)})) + (Z^{(j)} - M_n^{(j)}) \\ &\quad - \lambda_{n,q}^\top (Z^{(j)}(\mathbf{x}_{n+(1:q)}) - M_n^{(j)}(\mathbf{x}_{n+(1:q)})) \\ &\stackrel{\mathcal{D}}{=} Z^{(j)} + \lambda_{n,q}^\top (Z(\mathbf{x}_{n+(1:q)}) - Z^{(j)}(\mathbf{x}_{n+(1:q)})), \end{aligned}$$

which completes the proof. \square

Remark 0.1. It is actually sufficient that Z be Gaussian conditionally on $Z(\mathbf{x}_{1:n})$ for Proposition 2 to apply. This includes in particular the universal kriging case where the trend parameter is endowed with an improper uniform prior distribution.

In the next section, it is shown that the proposed algorithm has a lower complexity than the classical residual kriging algorithm, which would update the GRFs $Z^{(j)}$, $1 \leq j \leq J$ by computing kriging means based on all $n + q$ observations.

3 Complexity Calculation

3.1 Residual Kriging and FOXY Algorithms

This section details the computational complexity of simulating $J > 0$ sample paths of Z conditioned on $n + q$ observations at points $\mathbf{x}_{1:n}$, $\mathbf{x}_{n+(1:q)}$ in the case where J sample paths of Z conditioned on $Z(\mathbf{x}_{1:n})$ are already available. The results of the considered algorithms is an ensemble of J (conditionally) independent realizations of a GRF with conditional mean and covariance m_{n+q} and k_{n+q} given by the kriging equations. The algorithm is here given in the settings of universal kriging (Eqs. (10)-(12)). Only the case $n > 0$ will be discussed as, in case $n = 0$, the two detailed algorithms coincide.

Here, it is assumed that all simulations are performed at p simulation points $\mathbf{e}_1, \dots, \mathbf{e}_p$ in \mathbb{X} . To simplify complexity calculations it is further assumed that p, n are much larger than q . Another variable also assumed to be much smaller than p, n is the number ℓ of trend basis functions (Appendix A). To sum up, $p, n \gg q, \ell$.

Two major cases will be distinguished in the algorithms. First, an unfavorable case where the set of new observation points $\mathbf{x}_{n+(1:q)}$ is not included in the set of

simulation points $\{\mathbf{e}_1, \dots, \mathbf{e}_p\}$. Second, the favorable case where it is. Finally, two algorithms will be compared. The first one is a classical residual kriging algorithm, based on Eq. (1). This algorithm requires to compute a kriging mean M_{n+q} based on all the actual observations $Z(\mathbf{x}_{1:n}), Z(\mathbf{x}_{n+(1:q)})$ and also to obtain a kriging residual, which involves another computation of a kriging mean based on the observations $Z(\mathbf{x}_{1:n})$ and artificial observations $Z^{(j)}(\mathbf{x}_{n+(1:q)})$. These computations involve the calculation of $n+q$ kriging weights, which is done using Eq. (10). The kriging weights are the same for the two computed kriging means. The second algorithm is the FOXY algorithm which is based on Proposition 2 and Eq. (7). FOXY has the advantage of requiring only the computation of q kriging weights, $\lambda_{n,q}$, per simulation point. The number of kriging weights being reduced from $n+q$ to q , it is shown that FOXY brings a computational complexity reduction of $O(n/q)$ compared to the classical residual kriging algorithm. The complexities obtained will also be compared to the one of a third benchmark algorithm based on a decomposition (for example, Cholesky) of the conditional covariance matrix at the simulation points (Davis, 1987). This last algorithm does not take advantage of any previous computations. The algorithms are summarized in the next two subsections and details are given in Appendix B.

3.2 Preliminary Step: Generating $Z^{(j)}(\mathbf{x}_{n+(1:q)})$ for all J Sample Paths

An important detail that is relative to Eq. (1) and is even clearer in Eq. (7) is that updating J sample paths requires knowing, for each $j \in \{1, \dots, J\}$, the value of the sample path number j at batch $\mathbf{x}_{n+(1:q)}$, which is denoted by $Z^{(j)}(\mathbf{x}_{n+(1:q)})$. In Eq. (1) the knowledge of $Z^{(j)}(\mathbf{x}_{n+(1:q)})$ is required, as the computation of the kriging residual involves a kriging mean based on q artificial observations $Z^{(j)}(\mathbf{x}_{n+(1:q)})$. Thus, if the batch of q points $\mathbf{x}_{n+(1:q)}$ is not included in the set of p simulation points $\{\mathbf{e}_1, \dots, \mathbf{e}_p\}$, $Z^{(j)}(\mathbf{x}_{n+(1:q)})$ needs to be simulated conditionally on $n+p$ observations for all $j \in \{1, \dots, J\}$. This case is referred to as the unfavorable case. It involves the computation of the kriging weights in Eq. (10) of the $n+p$ points $\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{e}_1, \dots, \mathbf{e}_p$ for the prediction at each of the q points $\mathbf{x}_{n+(1:q)}$. In particular, the inversion of a $(n+p) \times (n+p)$ matrix in Eq. (10) is done with a cumbersome $O((n+p)^3)$ complexity. The cost for computing the other terms of Eq. (10) is dominated by this lower than the latter complexity. Finally, once the kriging weights are computed, $Z^{(j)}(\mathbf{x}_{n+(1:q)})$ is simulated for all j with a cost of $O(Jpq)$. Thus, in the described unfavorable case, both the classical residual kriging algorithm and the new algorithm based on Eq. (7) have a cost, referred to as the preliminary cost, which is of $O((n+p)^3 + Jpq)$. A detailed algorithm justifying this complexity is given in Appendix B. It will be shown that the preliminary cost tends to dominate the other costs of the two algorithms so that the savings provided by the FOXY algorithm are lower in the unfavorable case.

3.3 Computing Kriging Weights and Updating the Sample Paths

It is now assumed that, for all $j \in \{1, \dots, J\}$, $Z^{(j)}(\mathbf{x}_{n+(1:q)})$ is known. This is the case either if $\mathbf{x}_{n+(1:q)}$ is included in the set of simulation points $\{\mathbf{e}_1, \dots, \mathbf{e}_p\}$ (favorable case), or if the preliminary cost of $O((n+p)^3 + Jpq)$ has been paid (unfavorable case).

For the classical residual kriging algorithm, the computation of kriging means requires calculating $n+q$ kriging weights of $\mathbf{x}_1, \dots, \mathbf{x}_{n+q}$ for the prediction at points $\mathbf{e}_1, \dots, \mathbf{e}_p$. This will be done using Eq. (10). The inversion of the covariance matrix K at $n+q$ points in Eq. (10) has a $O((n+q)^3)$ complexity; and the p matrix-vector products $K^{-1}\mathbf{k}(\mathbf{x})$, where \mathbf{x} takes all the values $\mathbf{e}_1, \dots, \mathbf{e}_p$, have a $O(p(n+q)^2)$ cost. The computation of the other terms of Eq. (10) involves other complexities that are all dominated by the $O(p(n+q)^2)$ cost. The total cost is then of $O((n+q)^3 + p(n+q)^2)$. It is important to note that the $O((n+q)^3)$ complexity to invert K can be reduced to $O(qn^2)$ using matrix block-inversion formulae based on the Schur complement in the realistic case where the covariance matrix computed in the n (and not $n+q$) points $\mathbf{x}_1, \dots, \mathbf{x}_n$ has already been inverted or decomposed before. It shall be assumed that this is the case here, so that the dominating term in the complexity is now of $O(p(n+q)^2)$.

The FOXY algorithm based on Proposition 2 and Eq. (7) enables to improve this $O(p(n+q)^2)$ complexity, as shown below. It requires the computation of q kriging weights of $\mathbf{x}_{n+1}, \dots, \mathbf{x}_{n+q}$ for the prediction at points $\mathbf{e}_1, \dots, \mathbf{e}_p$. These weights are equal to $K_{n,q}^{-1}k_n(\mathbf{x}, \mathbf{x}_{n+(1:q)})$, as indicated by Eq. (6), where \mathbf{x} takes the values $\mathbf{e}_1, \dots, \mathbf{e}_p$, k_n is the kriging covariance defined in Eq. (12) and $K_{n,q} = k_n(\mathbf{x}_{n+(1:q)}, \mathbf{x}_{n+(1:q)})$ is the $q \times q$ kriging covariance matrix at $\mathbf{x}_{n+(1:q)}$ based on n observations. The computation of all these kriging covariances can be performed using Eq. (12). In that case, the cost to invert the matrix K is of $O(n^3)$ but, again, it is assumed that this inverse has already been computed. The remaining matrix-vector multiplications are performed at a cost of mainly $O((\ell+q)(n^2+pn))$. Complete details are given in Algorithm 4. This final cost is lower than the $O(p(n+q)^2)$ cost obtained with the classical residual kriging algorithm in the settings where $n, p \gg \ell, q$. In particular, for $q \geq \ell$ and $p \geq n$, FOXY has a complexity of $O(pnq)$ against $O(pn^2)$ for the residual kriging algorithm.

Once the kriging weights are computed, the remaining cost in both the residual kriging algorithm and FOXY is of $O(Jpq)$. This is simply the cost to perform q multiplications for all p simulation points $(\mathbf{e}_1, \dots, \mathbf{e}_p)$ and J sample paths. The aggregated complexity of the two studied algorithms are summarized in Table 3.3, together with the complexity of a standard algorithm based on a decomposition of the covariance matrix, which does not take advantage of previous computations. In the unfavorable case, FOXY brings a lower improvement with respect to the residual kriging algorithm, as the dominating terms in the complexity are $O((n+p)^3)$ and $O(Jpq)$ for both algorithms. On the other hand, FOXY is expected to be much faster than the classical residual kriging algorithm in the favorable case where $\mathbf{x}_{n+(1:q)}$ is

a subset of the set of simulation points, as the preliminary cost of $O((n+p)^3)$ is not paid.

Table 1 Theoretical complexity of the residual kriging algorithm (Eq. (1)), the FOXY algorithm based on Eq. (7) and an algorithm based on a decomposition of the covariance matrix at the observation points

algorithm	preliminary calc.	kriging weights	simulation
Residual kriging	$O((n+p)^3 + Jpq)$	$O(p(n+q)^2)$	$O(Jpq)$
FOXY	$O((n+p)^3 + Jpq)$	$O((\ell+q)(n^2 + pn))$	$O(Jpq)$
Decomposition-based			$O(p^3 + Jp^2)$

Remark 0.2. In FOXY, the computation of the q kriging weights $\lambda_{n,q}$ of Eq. (7) involves the computation of kriging covariances, as suggested by Eq. (6). In universal kriging, the kriging covariances are computed using Eq. (12), which requires to invert a $n \times n$ (and not $(n+q) \times (n+q)$) matrix. It was assumed that, for both algorithms (FOXY and residual kriging), this inverse was already computed. If this is not the case, the authors would recommend to be very cautious if appealing to matrix block-inversion formulae, especially if the GRF update formulae are applied recursively for more than one batch of q points. Numerical errors in the computation of the inverse of this matrix would indeed lead to errors in the kriging covariances and consequently in the kriging weights.

4 Numerical Experiments

This section illustrates the previously established complexity results with numerical experiments. The total computation time to update J simulations of a GRF conditioned on n observations into simulations conditioned on $n+q$ observations is measured. Simulations are performed at p simulation points. Here the simulated GRFs are indexed by $\mathbb{X} = [0, 1]^2$. The function used to condition the values of the GRFs is the rescaled Branin-Hoo function (?). Ordinary kriging is considered, which corresponds to the case of a single constant basis function ($\ell = 1$). As the computation times are not very sensitive to ℓ (at least if ℓ does not take large values), only the sensitivity of the computation time to J, n, p, q is investigated.

An example of GRFs simulated on a grid of $50 \times 50 = 2,500$ points is provided on Fig. 3. The three GRFs at the top are conditioned on $n = 10$ observations and each GRF is updated (using FOXY, see bottom plots) to be conditioned on $10 + 3$ observations. For the experiments, a set of values needs to be chosen for (J, n, p, q) . The chosen set is the following full factorial design,

$$(J, n, p, q) \in \mathbb{D}_J \times \mathbb{D}_n \times \mathbb{D}_p \times \mathbb{D}_q,$$

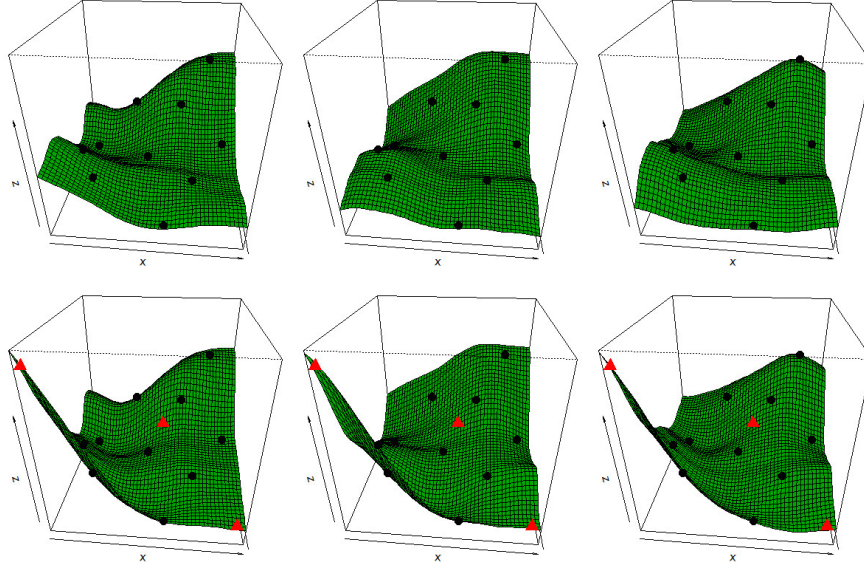


Fig. 3 Top: three realizations of a GRF in two dimensions conditioned on $n = 10$ observations (black points). Bottom: update of these three realizations when $q = 3$ new observations (red triangles) are assimilated

where $\mathbb{D}_J = \{1; 100; 1,000; 10,000; 20,000; 30,000\}$, $\mathbb{D}_n = \{10; 100; 500; 1,000\}$, $\mathbb{D}_p = \{100; 500; 2,000\}$ and $\mathbb{D}_q = \{1; 10\}$. The set \mathbb{D}_J covers cases where very few conditional simulations are updated, so that the computation time will be dominated by the preliminary costs or by the time to compute kriging weights. \mathbb{D}_J also covers cases where a large set of conditional simulations is updated. Regarding \mathbb{D}_n , the number n of initial conditioning observations goes from very few (10) to up to 1,000. The reason for not going beyond a value of 1,000 is that standard uses of kriging with a unique neighborhood are rarely done when the number of observations is larger because of a necessary $n \times n$ matrix inversion. For the same reason the value of p does not go beyond 2,000. It must be recalled that in the unfavorable case, a $(n + p) \times (n + p)$ matrix needs to be inverted. One may however note that in the favorable case it is perfectly possible to use large p , or even infinite p . Finally, only two small values of q were considered as the change of computation time when q grows is rather simple for moderate q . For all possible (n, p, q) triplets the computation time is computed as a function of J . Experiments were performed in both the unfavorable and favorable cases on a laptop with a 2.27 Ghz cpu and 3.7 Gb of RAM.

As explained in the previous section, the total computation time of the tested algorithms is the sum of the computation times of three different steps:

1. The preliminary cost (unfavorable case). Here both algorithms perform a $(n + p) \times (n + p)$ matrix inversion, which adds a fixed cost that does not depend on J .
2. The cost to compute kriging weights. In this step the FOXY algorithm computes q kriging weights while the residual kriging algorithm computes $n + q$ weights. This cost is also a fixed cost that does not depend on J . This step is where FOXY might be much faster than the classical algorithm.
3. The cost to update simulations once the weights are computed. Both algorithms have an $O(Jpq)$ cost for this step. Hence, when q is large, computation time is expected to grow faster with J .

Figure 4 details the results in the favorable case while Fig. 5 shows the unfavorable case. Computation times are given in seconds. It is important to note that, for a given value of (n, p, q) , the improvement brought by FOXY, expressed in number of seconds, is the same in the favorable and unfavorable case. However, the ratio between the computation times of the two algorithms does not remain the same, as the preliminary costs can be important. This explains why even if the time difference between the two methods is unchanged, the computation times of the two algorithms, plotted on a log-scale, seem to be closer in the unfavorable case.

The following conclusions can be drawn from Figs 4, 5. First, whenever n is very low, the time to compute kriging weights becomes negligible, which explains why the two algorithms have the same performances. In that case, the computation time is dominated by the $O(Jpq)$ complexity to update the J conditional simulations, or, in the unfavorable case, by the preliminary costs.

When $n = 100$, the computation time to obtain the kriging weights is not negligible anymore. In the favorable case, FOXY reduces the computation time by a factor two for moderate (less than 1,000) J . For larger J the performances tend to be similar as the $O(Jpq)$ complexity dominates again. In the unfavorable case, FOXY reduces the computation time by a factor which is lower than two because of the preliminary costs paid in both algorithms.

Finally, whenever $n = 500$ or 1,000 the gap between the two algorithms is large. This is due to the time to compute the kriging weights that is $O(n/q)$ faster with FOXY. For $n = 1,000$ and in the favorable case, FOXY can be up to 25 times faster for moderate J and 10 times faster for large J . In the unfavorable case, the improvements are less substantial and essentially depend on p instead of J . For low p , FOXY is approximately three times faster, and only two times faster for large p .

The choice of q mainly influences the cost of the last step of both algorithms, which has a $O(Jpq)$ complexity. q is thus driving the slope of the curves presented on Figs. 4, 5. It is also interesting to note that, in absolute value, computation times are generally low. With a very standard laptop, the update of 30,000 simulations on 2,000 points conditionally on 1,000 observations takes only one second with FOXY in the favorable case, and ten seconds in the unfavorable case. Performing the simulation from scratch using a Cholesky decomposition takes approximately 20 minutes, here.

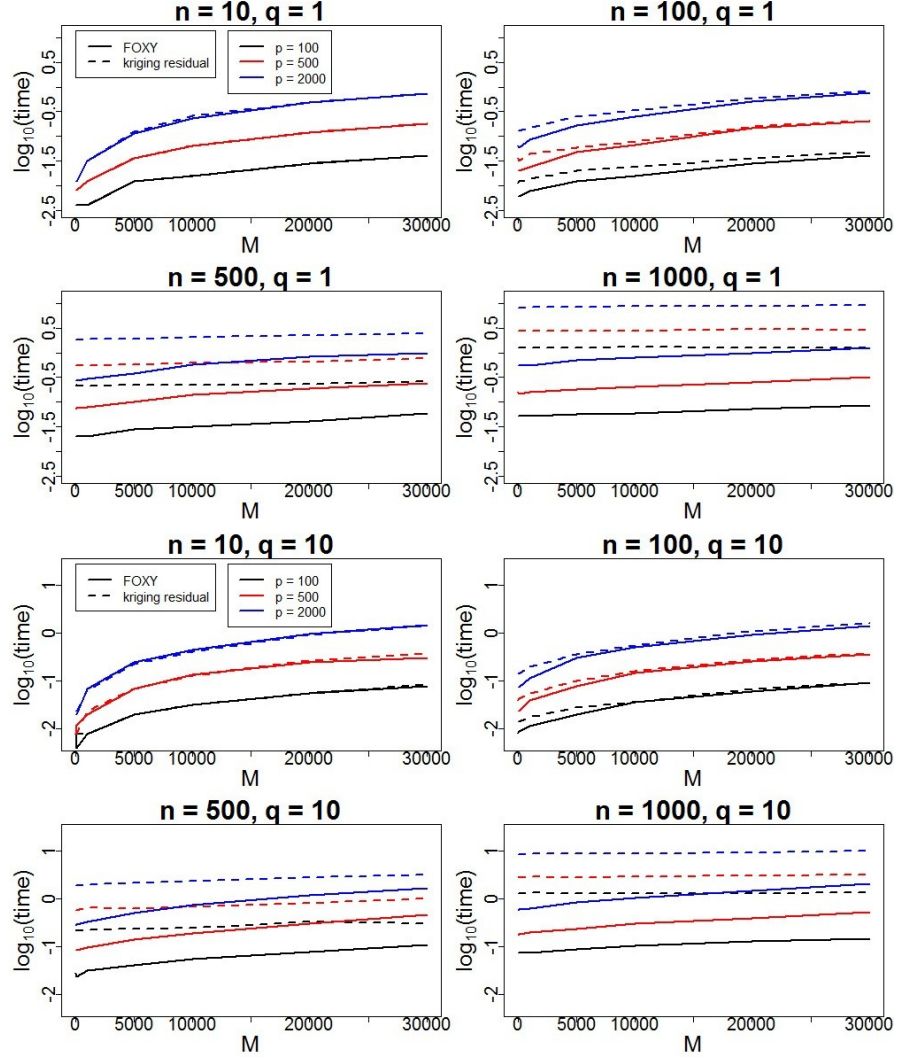


Fig. 4 Computation times in the favorable case

5 Conclusion

This paper presents efficient formulae allowing to quickly update ensembles of simulations of GRFs. Simulated paths, which are initially conditioned on $n \geq 0$ observations, are twisted in order to be conditioned on $n + q$ observations, for arbitrary $q \geq 1$. The formulae lead to a fast update algorithm that has been implemented in R and proves to offer substantial computational savings, especially when the number of conditioning observations n is large. In addition, the formulae have the advan-

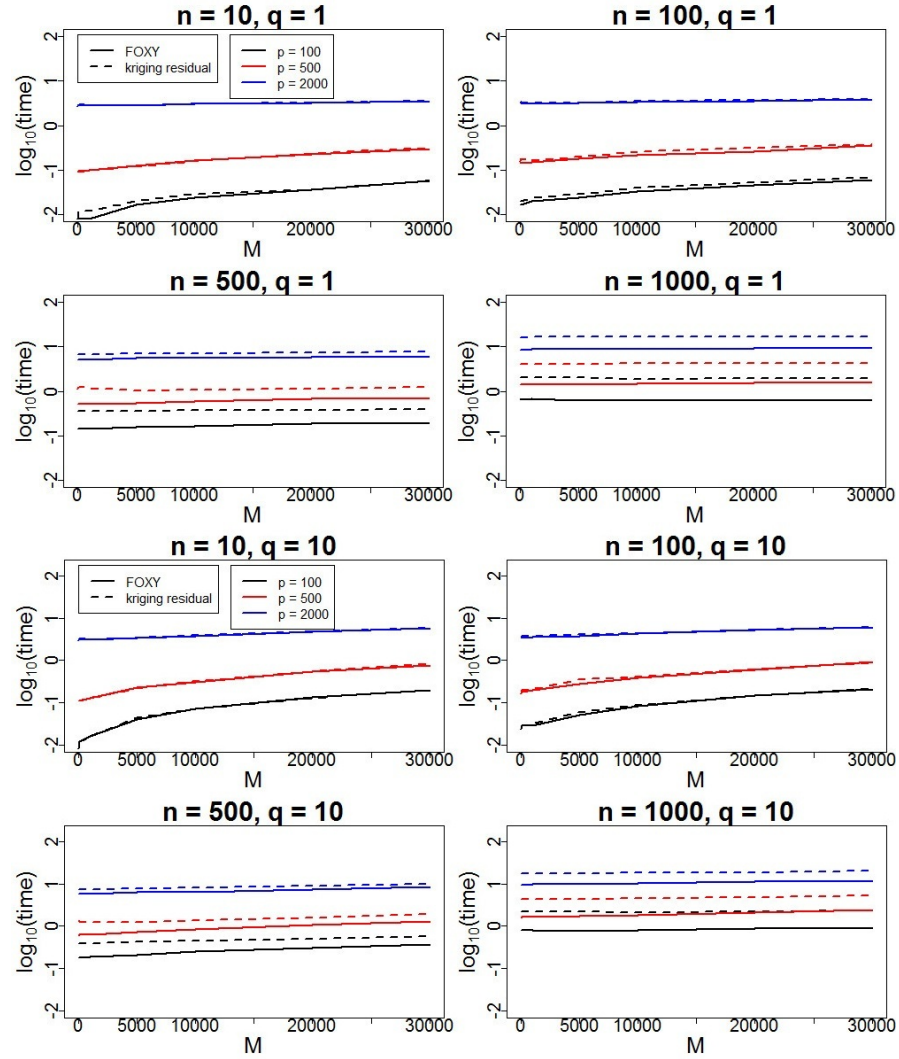


Fig. 5 Computation times in the unfavorable case

tage of explicitly quantifying the effect of the q newly assimilated observations on already simulated sample paths.

A limitation of the formulae, though, is that they apply only in the case where the covariance parameters of the non-conditional covariance function of Z are assumed known. In the typical settings where the covariance parameters are re-estimated when new observations are assimilated, the formulae cannot be straightforwardly applied.

The fast update approaches presented in this paper can be applied to efficiently compute Monte-Carlo estimates based on GRF simulations in the case where observations are assimilated sequentially. Example of potential applications include estimating the Shannon entropy of the maximizer of a conditioned GRF, a crucial step in a recently proposed Bayesian global optimization algorithm (Villemonteix et al., 2009). In the same vein, updated simulations have been recently used in the framework of the robust inversion problem studied in Chevalier (2013). In any case, when relying on an ensemble conditional simulations, one has to keep in mind that potential biases in Monte-Carlo estimates due to sample finiteness may propagate along consecutive stages. Hence, when using such an approach, and especially when it comes to uncertainty quantification purposes, the sample size and the procedure for generating the initial set of simulations should be carefully chosen.

Acknowledgements Part of this work has been conducted within the frame of the ReDice Consortium, gathering industrial (CEA, EDF, IFPEN, IRSN, Renault) and academic (Ecole des Mines de Saint-Etienne, INRIA, and the University of Bern) partners around advanced methods for Computer Experiments. Clément Chevalier warmly thanks Prof. Julien Bect for fruitful discussions on GRF simulation. David Ginsbourger acknowledges support from the Department of Mathematics and Statistics of the University of Bern and from the Integrated methods for stochastic ensemble aquifer modelling (ENSEMBLE) project funded by the Swiss National Science Foundation under the contract CRSI22.12249/1. The authors are indebted to two anonymous referees, an associate editor, and Yves Deville for a number of remarks having contributed to substantially improve the paper.

References

- Barnes, R. J. and Watson, A. (1992). Efficient updating of kriging estimates and variances. *Mathematical geology*, 24(1):129–133.
- Binois, M., Ginsbourger, D., and Roustant, O. (2014). Quantifying uncertainty on pareto fronts with gaussian process conditional simulations. Technical report, ReDICE Consortium.
- Chevalier, C. (2013). *Fast Uncertainty reduction strategies relying on Gaussian process models*. PhD thesis, University of Bern.
- Chevalier, C., Ginsbourger, D., and Emery, X. (2014). Corrected kriging update formulae for batch-sequential data assimilation. In *Mathematics of Planet Earth*, pages 119–122. Springer.
- Chilès, J. and Allard, D. (2005). Stochastic simulation of soil variations. In Grunwald, S., editor, *Environmental Soil-Landscape Modeling: Geographic Information Technologies and Pedometrics*, pages 289–321. CRC Press.
- Chilès, J.-P. and Delfiner, P. (2012). *Geostatistics: Modeling Spatial Uncertainty*. Wiley, New York, second edition edition.
- Davis, M. W. (1987). Production of conditional simulations via the lu triangular decomposition of the covariance matrix. *Mathematical Geology*, 19(2):91–98.
- de Fouquet, C. (1994). Reminders on the conditioning kriging. In Armstrong, M. and Dowd, P., editors, *Geostatistical Simulations*, pages 131–145. Kluwer Academic Publishers.
- Delhomme, J. (1979). Spatial variability and uncertainty in groundwater flow parameters: a geostatistical approach. *Water Resources Research*, 15(2):269–280.
- Deutsch, C. (2002). *Geostatistical Reservoir Modeling*. Oxford University Press, New York.
- Dimitrakopoulos, R. (2011). *Advances in Ore Body Modelling and Strategic Mine Planning*. Australasian Institute of Mining and Metallurgy, Melbourne.

- Emery, X. (2009). The kriging update equations and their application to the selection of neighboring data. *Computational Geosciences*, 13(1):211–219.
- Emery, X. and Lantu  joul, C. (2006). Tbsim: A computer program for conditional simulation of three-dimensional gaussian random fields via the turning bands method. *Computers & Geosciences*, 32(10):1615–1628.
- Gao, H., Wang, J., and Zhao, P. (1996). The updated kriging variance and optimal sample design. *Mathematical Geology*, 28:295–313.
- Handcock, M. S. and Stein, M. L. (1993). A bayesian analysis of kriging. *Technometrics*, 35(4):403–410.
- Hern  ndez, J. and Emery, X. (2009). A geostatistical approach to optimize sampling designs for local forest inventories. *Canadian Journal of Forest Research*, 39:1465–1474.
- Hoshiya, M. (1995). Kriging and conditional simulation of gaussian field. *Journal of engineering mechanics*, 121 (2):181–186.
- Journ  l, A. and Kyriakidis, P. (2004). *Evaluation of Mineral Reserves: a Simulation Approach*. Oxford University Press, New York.
- Matheron, G. (1973). The intrinsic random functions and their applications. *Advances in applied probability*, pages 439–468.
- O’Hagan, A. (1978). Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, 40(1):1–42.
- Omre, H. and Halvorsen, K. (1989). The bayesian bridge between simple and universal kriging. *Mathematical Geology*, 22 (7):767–786.
- Roustant, O., Ginsbourger, D., and Deville, Y. (2012). Dicekriging, Diceoptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. *Journal of Statistical Software*, 51.
- Santner, T. J., Williams, B. J., and Notz, W. (2003). *The Design and Analysis of Computer Experiments*. Springer Verlag.
- Stein, M. L. (1999). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer, New York.
- Villemonteix, J., Vazquez, E., and Walter, E. (2009). An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44(4):509–534.

Appendix A: Universal and Simple Kriging

Let Z be a L^2 random field defined on a bounded set $\mathbb{X} \subset \mathbb{R}^d$ with known (and not necessarily stationary) covariance function $k(\cdot, \cdot)$ and unknown mean function $m(\cdot)$ such that $Z|m \sim \text{GRF}(m, k)$, where $\text{GRF}(m, k)$ denotes a GRF with mean function m and covariance function k . A well-known Bayesian approach consists in writing m as follows

$$m(\cdot) = \sum_{i=1}^{\ell} \beta_i f_i(\cdot),$$

where $\ell \geq 1$, f_1, \dots, f_ℓ are ℓ known basis functions and $\beta = (\beta_1, \dots, \beta_\ell)$ has an improper uniform prior in \mathbb{R}^ℓ . In these settings, known as the universal kriging settings, when n observations $Z(\mathbf{x}_{1:n})$ are assimilated at points $\mathbf{x}_{1:n} := (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathbb{X}^n$, it is known (O’Hagan, 1978) that the posterior distribution of Z is a GRF with posterior (or conditional) mean function m_n^{UK} and covariance function k_n^{UK} given by the so-called universal kriging equations

$$\lambda^{UK}(\mathbf{x}) = K^{-1} \left(\mathbf{k}(\mathbf{x}) + \mathbb{F}(\mathbb{F}^\top K^{-1} \mathbb{F})^{-1} (\mathbf{f}(\mathbf{x}) - \mathbb{F}^\top K^{-1} \mathbf{k}(\mathbf{x})) \right), \quad (10)$$

$$m_n^{UK}(\mathbf{x}) = \lambda^{UK}(\mathbf{x})^\top Z(\mathbf{x}_{1:n}) = \mathbf{f}(\mathbf{x})^\top \hat{\boldsymbol{\beta}} + \mathbf{k}(\mathbf{x})^\top K^{-1} \left(Z(\mathbf{x}_{1:n}) - \mathbb{F} \hat{\boldsymbol{\beta}} \right), \quad (11)$$

$$k_n^{UK}(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - \mathbf{k}(\mathbf{x})^\top K^{-1} \mathbf{k}(\mathbf{x}') + (\mathbf{f}(\mathbf{x})^\top - \mathbf{k}(\mathbf{x})^\top K^{-1} \mathbb{F})(\mathbb{F}^\top K^{-1} \mathbb{F})^{-1} (\mathbf{f}(\mathbf{x}')^\top - \mathbf{k}(\mathbf{x}')^\top K^{-1} \mathbb{F})^\top, \quad (12)$$

where $\hat{\boldsymbol{\beta}} := (\mathbb{F}^\top K^{-1} \mathbb{F})^{-1} \mathbb{F}^\top K^{-1} Z(\mathbf{x}_{1:n})$, $\mathbf{f}(\mathbf{x}) := (f_1(\mathbf{x}), \dots, f_\ell(\mathbf{x}))^\top$, $\mathbb{F} \in \mathbb{R}^{n \times \ell}$ is the matrix with row i equal to $\mathbf{f}(\mathbf{x}_i)^\top$, $\mathbf{k}(\mathbf{x}) := (k(\mathbf{x}, \mathbf{x}_1), \dots, k(\mathbf{x}, \mathbf{x}_n))^\top$, K is the covariance matrix at the observation points, $K := (k(\mathbf{x}_i, \mathbf{x}_j))_{1 \leq i, j \leq n}$. The vector $\lambda^{UK}(\mathbf{x})$ is the vector of n kriging weights of $\mathbf{x}_1, \dots, \mathbf{x}_n$ for the prediction at point \mathbf{x} .

A well-known simpler setting is the case where the non-conditional mean function m is already known. In that case, the Bayesian approach is no longer necessary and the conditional mean and covariance function of Z are given by the so-called simple kriging equations, written here in the case where $m(\cdot) = 0$

$$\lambda^{SK}(\mathbf{x}) = K^{-1} \mathbf{k}(\mathbf{x}), \quad (13)$$

$$m_n^{SK}(\mathbf{x}) = \lambda^{SK}(\mathbf{x})^\top Z(\mathbf{x}_{1:n}) = \mathbf{k}(\mathbf{x})^\top K^{-1} Z(\mathbf{x}_{1:n}), \quad (14)$$

$$k_n^{SK}(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - \mathbf{k}(\mathbf{x})^\top K^{-1} \mathbf{k}(\mathbf{x}'). \quad (15)$$

If m is not equal to zero, the simple kriging covariance function k_n^{SK} is unchanged and an application of Eq. (14) to the centred GRF $Z - m$ yields $m_n^{SK}(\mathbf{x}) = m(\mathbf{x}) + \mathbf{k}(\mathbf{x})^\top K^{-1} (Z(\mathbf{x}_{1:n}) - m(\mathbf{x}_{1:n}))$.

Appendix B: Algorithms

Algorithm 2 (Preliminary cost): computation of $Z^{(j)}(\mathbf{x}_{n+(1:q)})$ for all $1 \leq j \leq J$.

Require: J i.i.d. GRFs $Z^{(1)}, \dots, Z^{(J)}$ simulated in p points $\mathbf{e}_1, \dots, \mathbf{e}_p$ conditionally on $n \geq 0$ observations $Z(\mathbf{x}_1), \dots, Z(\mathbf{x}_n)$ at points $\mathbf{x}_1, \dots, \mathbf{x}_n$; and $q > 0$ additional points $\mathbf{x}_{n+(1:q)} = (\mathbf{x}_{n+1}, \dots, \mathbf{x}_{n+q})$.

Require: The non-conditional covariance function, k , of the GRFs $Z^{(1)}, \dots, Z^{(J)}$.

Step 1. The goal is to compute a matrix of $(n+p) \times q$ kriging weights of $(\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{e}_1, \dots, \mathbf{e}_p)$ for the prediction at points $\mathbf{x}_{n+1}, \dots, \mathbf{x}_{n+q}$, using Eq. (10).

- Compute the inverse of the covariance matrix at points $(\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{e}_1, \dots, \mathbf{e}_p)$: $\mathbf{O}((\mathbf{n} + \mathbf{p})^3)$

- Compute other terms of Eq. (10) which do not depend on \mathbf{x} : $\mathbf{O}(\ell(\mathbf{n} + \mathbf{p})^2 + \ell^2(\mathbf{n} + \mathbf{p}) + \ell^3)$

for $i = 1 \rightarrow q$ **do**

- Compute $\mathbf{k}(\mathbf{x}_{n+i}) = (k(\mathbf{x}_{n+i}, \mathbf{x}_1), \dots, k(\mathbf{x}_{n+i}, \mathbf{x}_n), k(\mathbf{x}_{n+i}, \mathbf{e}_1), \dots, k(\mathbf{x}_{n+i}, \mathbf{e}_p))^\top$ and $\mathbf{f}(\mathbf{x}_{n+i})$.

- Compute the multiplication $K^{-1} \mathbf{k}(\mathbf{x}_{n+i})$ and then $\mathbb{F}^\top K^{-1} \mathbf{k}(\mathbf{x}_{n+i})$. Conclude the computation of $\lambda^{UK}(\mathbf{x}_{n+i})$: $\mathbf{O}(\ell(\mathbf{n} + \mathbf{p}) + (\mathbf{n} + \mathbf{p})^2)$

end for

Total cost for Step 1 dominated by the term of $\mathbf{O}((\mathbf{n} + \mathbf{p})^3)$

Step 2. Preliminary: compute $S := (k_{n+p}(\mathbf{x}_{n+i}, \mathbf{x}_{n+j}))_{1 \leq i, j \leq q}$ using Eq. (12) and the terms precomputed in Step 1. Compute also a decomposition (Cholesky, Mahalanobis) of S : $\mathbf{O}(q^2(\mathbf{n} + \mathbf{p}) + q^3)$.

for $i = 1 \rightarrow q$ **do**

- If $n > 0$, precompute, $u_i := \sum_{j=1}^n [\lambda^{UK}(\mathbf{x}_{n+i})]_j Z(\mathbf{x}_j)$, where $[\lambda^{UK}(\mathbf{x}_{n+i})]_j$ is the kriging weight of \mathbf{x}_j for the prediction at point \mathbf{x}_{n+i} : $\mathbf{O}(\mathbf{n})$

end for

for $j = 1 \rightarrow J$ **do**

for $i = 1 \rightarrow q$ **do**

- Compute the kriging mean $m_{n+p}^{UK}(\mathbf{x}_{n+i})$ from the $n + p$ observations $Z(\mathbf{x}_1), \dots, Z(\mathbf{x}_n), Z^{(j)}(\mathbf{e}_1), \dots, Z^{(j)}(\mathbf{e}_p)$, using the vector of kriging weights $\lambda^{UK}(\mathbf{x}_{n+i})$. The precomputation of u_i reduces this to p operations. $\mathbf{O}(\mathbf{p})$.

end for

- Simulate a Gaussian random vector $Z^{(j)}(\mathbf{X}_q)$ with mean $m_{n+p}^{UK}(\mathbf{X}_q)$ and covariance matrix S . $\mathbf{O}(q^2)$

end for

Total cost for Step 2 dominated by the term of $\mathbf{O}(\mathbf{Jp}q)$

Total cost of the algorithm dominated by the term of $\mathbf{O}(\mathbf{Jp}q + (\mathbf{n} + \mathbf{p})^3)$

Algorithm 3 Classical residual kriging algorithm

Require: J i.i.d. GRFs $Z^{(1)}, \dots, Z^{(J)}$ simulated in $p+q$ points $\mathbf{e}_1, \dots, \mathbf{e}_p, \mathbf{x}_{n+1}, \dots, \mathbf{x}_{n+q}$ conditionally on $n \geq 0$ observations $\mathbf{x}_1, \dots, \mathbf{x}_n$.

Require: The non-conditional covariance function, k , of the GRFs $Z^{(1)}, \dots, Z^{(J)}$.

Require: $q > 0$ real observations $Z(\mathbf{x}_{n+(1:q)}) = (Z(\mathbf{x}_{n+1}), \dots, Z(\mathbf{x}_{n+q}))$

Require: If $n > 0$, the inverse of the matrix $K_0 := (k(\mathbf{x}_i, \mathbf{x}_j))_{1 \leq i, j \leq n}$

Step 1. The goal is to compute $(n+q) \times p$ kriging weights of $(\mathbf{x}_1, \dots, \mathbf{x}_{n+q})$ for the prediction at points $\mathbf{e}_1, \dots, \mathbf{e}_p$, using Eq. (10). These weights will be used to obtain kriging means at points $\mathbf{e}_1, \dots, \mathbf{e}_p$.

- Inverse the covariance matrix K at points $(\mathbf{x}_1, \dots, \mathbf{x}_{n+q})$ with block-inversion formula: $\mathbf{O}(\mathbf{q}\mathbf{n}^2)$.

- Compute the other terms in Eq. (10) which do not depend on \mathbf{x} : $\mathbf{O}(\ell(\mathbf{n}+\mathbf{q})^2 + \ell^2(\mathbf{n}+\mathbf{q}) + \ell^3)$

for $i = 1 \rightarrow p$ **do**

- Compute $\mathbf{k}(\mathbf{e}_i) = (k(\mathbf{e}_i, \mathbf{x}_1), \dots, k(\mathbf{e}_i, \mathbf{x}_{n+q}))^\top$ and $\mathbf{f}(\mathbf{e}_i)$: $\mathbf{O}(\mathbf{n}+\mathbf{q} + \ell)$

- Compute $K^{-1}\mathbf{k}(\mathbf{e}_i)$, then $\mathbb{F}^\top K^{-1}\mathbf{k}(\mathbf{e}_i)$, then $\lambda^{UK}(\mathbf{e}_i)$: $\mathbf{O}(\ell(\mathbf{n}+\mathbf{q}) + (\mathbf{n}+\mathbf{q})^2)$

end for

Total cost for Step 1 dominated by the term of $\mathbf{O}(\mathbf{p}(\mathbf{n}+\mathbf{q})^2)$

Step 2. The goal is to use the kriging weights to compute, for all the J GRF sample paths, kriging means at points $\mathbf{e}_1, \dots, \mathbf{e}_p$ based on $n+q$ observations at points $\mathbf{x}_1, \dots, \mathbf{x}_{n+q}$.

for $i = 1 \rightarrow p$ **do**

- If $n > 0$, precompute, $u_i := \sum_{j=1}^n [\lambda^{UK}(\mathbf{e}_i)]_j Z(\mathbf{x}_j)$, where $[\lambda^{UK}(\mathbf{e}_i)]_j$ is the kriging weight of \mathbf{x}_j for the prediction at point \mathbf{e}_i : $\mathbf{O}(\mathbf{n})$

- Compute the kriging mean at point \mathbf{e}_i from $Z(\mathbf{x}_1), \dots, Z(\mathbf{x}_n), Z(\mathbf{x}_{n+1}), \dots, Z(\mathbf{x}_{n+q})$, using the vector of kriging weights $\lambda^{UK}(\mathbf{e}_i)$. The precomputation of u_i reduces this to q operations. $\mathbf{O}(\mathbf{q})$.

end for

for $j = 1 \rightarrow J$ **do**

for $i = 1 \rightarrow p$ **do**

- Compute the kriging mean at point \mathbf{e}_i from $Z(\mathbf{x}_1), \dots, Z(\mathbf{x}_n), Z^{(j)}(\mathbf{x}_{n+1}), \dots, Z^{(j)}(\mathbf{x}_{n+q})$, and using precomputations. Apply Eq. (1) to obtain $Z^{*(j)}(\mathbf{e}_i)$: $\mathbf{O}(\mathbf{q})$.

end for

end for

Total cost for Step 2: $\mathbf{O}(\mathbf{J}\mathbf{p}\mathbf{q})$

Total cost of the algorithm dominated by the term of $\mathbf{O}(\mathbf{J}\mathbf{p}\mathbf{q} + \mathbf{p}(\mathbf{n}+\mathbf{q})^2)$

Algorithm 4 FOXY algorithm based on Proposition 2

Require: J i.i.d. GRFs $Z^{(1)}, \dots, Z^{(J)}$ simulated in $p+q$ points $\mathbf{e}_1, \dots, \mathbf{e}_p, \mathbf{x}_{n+1}, \dots, \mathbf{x}_{n+q}$ conditionally on $n \geq 0$ observations $\mathbf{x}_1, \dots, \mathbf{x}_n$.

Require: The non-conditional covariance function, k , of the GRFs $Z^{(1)}, \dots, Z^{(J)}$.

Require: $q > 0$ real observations $Z(\mathbf{x}_{n+(1:q)}) = (Z(\mathbf{x}_{n+1}), \dots, Z(\mathbf{x}_{n+q}))$

Require: If $n > 0$, the inverse of the matrix $K := (k(\mathbf{x}_i, \mathbf{x}_j))_{1 \leq i, j \leq n}$
 Step 1. The goal is to compute the q kriging weights $\lambda_{n,q}(\mathbf{x}) = K_{n,q}^{-1} k_n^{UK}(\mathbf{x}, \mathbf{X}_q)$ of Eq. (7) where \mathbf{x} takes the values $\mathbf{e}_1, \dots, \mathbf{e}_p$, $K_{n,q} = (k_n^{UK}(\mathbf{x}_{n+i}, \mathbf{x}_{n+j}))_{1 \leq i, j \leq q}$ and k_n^{UK} is obtained using Eq. (12).

- Compute terms of Eq. (12) which do not depend on \mathbf{x} : $\mathbf{O}(\ell \mathbf{n}^2 + \ell^2 \mathbf{n} + \ell^3)$

for $i = 1 \rightarrow p$ **do**

- Compute $\mathbf{k}(\mathbf{e}_i) = (k(\mathbf{e}_i, \mathbf{x}_1), \dots, k(\mathbf{e}_i, \mathbf{x}_n))^T$ and $\mathbf{f}(\mathbf{e}_i)$: $\mathbf{O}(\mathbf{n} + \ell)$

- Compute $\mathbf{k}(\mathbf{e}_i)^T (K^{-1} \mathbb{F})$, then $\mathbf{f}(\mathbf{e}_i)^T - \mathbf{k}(\mathbf{e}_i)^T (K^{-1} \mathbb{F})$: $\mathbf{O}(\ell \mathbf{n})$

end for

for $i = 1 \rightarrow q$ **do**

- Compute $\mathbf{k}(\mathbf{x}_{n+i}) = (k(\mathbf{x}_{n+i}, \mathbf{x}_1), \dots, k(\mathbf{x}_{n+i}, \mathbf{x}_n))^T$ and $\mathbf{f}(\mathbf{x}_{n+i})$: $\mathbf{O}(\mathbf{n} + \ell)$

- Compute $\mathbf{k}(\mathbf{x}_{n+i})^T (K^{-1} \mathbb{F})$, then $\mathbf{f}(\mathbf{x}_{n+i})^T - \mathbf{k}(\mathbf{x}_{n+i})^T (K^{-1} \mathbb{F})$: $\mathbf{O}(\ell \mathbf{n})$

- Compute $K^{-1} \mathbf{k}(\mathbf{x}_{n+i})$, then $(\mathbb{F}^T K^{-1} \mathbb{F})^{-1} (\mathbf{f}(\mathbf{x}_{n+i})^T - \mathbf{k}(\mathbf{x}_{n+i})^T K^{-1} \mathbb{F})^T$: $\mathbf{O}(\mathbf{n}^2 + \ell^2)$

end for

- Conclude the calculation of $k_n^{UK}(\mathbf{x}, \mathbf{x}')$ for all $(\mathbf{x}, \mathbf{x}') \in \mathbf{X}_q \times \mathbf{X}_q$: $\mathbf{O}(\mathbf{q}^2(\mathbf{n} + \ell))$

- Conclude the calculation of $k_n^{UK}(\mathbf{x}, \mathbf{x}')$ for all $(\mathbf{x}, \mathbf{x}') \in \{\mathbf{e}_1, \dots, \mathbf{e}_p\} \times \mathbf{X}_q$: $\mathbf{O}(\mathbf{p}\mathbf{q}(\mathbf{n} + \ell))$

- Compute kriging weights through $k_n^{UK}(\mathbf{x}, \mathbf{X}_q)^T K_n^{-1}$ for all $\mathbf{x} \in \{\mathbf{e}_1, \dots, \mathbf{e}_p\}$: $\mathbf{O}(\mathbf{q}^3 + \mathbf{p}\mathbf{q}^2)$

Total cost for Step 1 dominated by the term of $\mathbf{O}((\mathbf{p}\mathbf{n} + \mathbf{n}^2)(\ell + \mathbf{q}))$

Step 2. Conclude the update of the J GRFs by applying Eq. (7). For a given point \mathbf{x} and a given GRF sample path, q operations are performed. Total cost of $\mathbf{O}(\mathbf{J}\mathbf{p}\mathbf{q})$.

Total cost of the algorithm dominated by the term of $\mathbf{O}(\mathbf{J}\mathbf{p}\mathbf{q} + (\mathbf{p}\mathbf{n} + \mathbf{n}^2)(\ell + \mathbf{q}))$